Dummy Data Generation and Data Augmentation using Machine Learning

Omar Bin Sheik Mustafa

School of Electrical & Electronic Engineering
Nanyang Technological University, Singapore
50 Nanyang Ave.
E-mail: omar004@e.ntu.edu.sg

Abstract: This paper gives an overview of the challenges associated with the scarcity of data and the impacts of limited data in machine learning applications. For this study, we examine the concept of data augmentation and explore the use of Generative Adversarial Networks (GANs) and Neural Style Transfer (NST) in generating synthetic data to overcome these limitations. The paper first goes into detail about the reasons behind data scarcity and its consequences on machine learning model accuracy and performance. We then investigate various data augmentation techniques that can effectively augment current existing datasets, before exploring the capabilities of GANs and NST as efficient approaches for generating synthetic data. These techniques help to address data scarcity issues and improve the performance of the machine learning model. In the conclusion of the paper, we analyse the benefits and limitations of dummy data generation, and its potential usage in future research and work.

Keywords: Data Augmentation, Machine Learning, Generative Adversarial Networks, Neural Style Transfer

1. Introduction

Machine learning, a subfield of artificial intelligence, concentrates on creating algorithms that enable software applications to enhance their predictive accuracy without explicit programming (Burns, 2019). It is based on statistical and mathematical principles and utilises various techniques such as supervised and unsupervised learning (MathWorks, n.d.). The emergence of artificial intelligence and machine learning has revolutionised various fields of technology by allowing machines to learn and make precise predictions from data. Machine learning is used every day in the real-world, with some examples of applications being in facial recognition, social media optimisation, and predictive analysis (Tableau, n.d.). One crucial factor in the accuracy of machine learning models is the availability of large, diverse, and high-quality datasets. However, obtaining such datasets can often be time-consuming, expensive, and impractical (JavaTpoint, n.d.). The accuracy of machine learning models depends on the quantity of training data available. The more training data the model has, the more accurate it will be. Consequently, obtaining an adequate amount of data for training is crucial for harnessing the power of machine learning effectively.

2. Literature Review

2.1 Scarcity of data

Contrary to the natural assumption that nearly every business or market is inundated with a deluge of data, the reality is that data is often scarce and only accessible to a select group of companies, added with the uncertain quality of any readily available data (Abadi, 2021). About 96% of enterprises encounter data quality and labelling challenges when it comes to machine learning projects (Dimensional Research, 2019). Acquiring large quantities of accurately labelled data is difficult due to reasons such as high costs, time, and effort needed for data collection or annotation. Crowdsourcing is a possible method to alleviate human labour, however, this approach can often lead to low-quality results (Zhou, 2017). Privacy and security concerns also impede data acquisition, especially in areas like big data analytics. To ensure the protection of privacy, various mechanisms have been developed and put in place for data generation, storage, and processing phase. These include access restrictions, data falsification, encryption, and anonymisation techniques (Jain, 2016). Consequently, these measures to protect data privacy contribute to the difficulties in the acquisition of data.

2.2 Impact of limited data

Overfitting and underfitting are significant issues in machine learning. Overfitting occurs when an algorithm adapts to the training dataset so precisely that it memorises the noise and unique characteristics of that data. Conversely, underfitting occurs when the model is unable to encompass the variability present in the data. This leads to high error rates on training and on unfamiliar data, making the model unsuitable for classification or prediction tasks. The ability of a model to generalise new data is ultimately what enables us to utilise machine learning algorithms every day to make predictions and categorise data (IBM, n.d.) (Jabbar & Khan, 2015).

2.3 Data requisites

The amount of data needed to train a machine learning model depends on variables like model and learning algorithm complexity, labelling needs, acceptable error margin, and input diversity. Larger datasets reduce biasness and randomness, resulting in the creation of more robust models. A common way to determine if a dataset is adequate is to apply the 10 times rule, or rule-of-thumb approach, which suggests that the number of examples should be at least 10 times greater than the degrees of freedom of the model. It is crucial to have appropriate coverage across various categories or classes within a dataset to avoid class imbalance or sampling bias issues (Dorfman, 2022) (Smolic, 2022).

2.4 Dummy data generation and data augmentation

2.4.1 Dummy data

Dummy data, also known as synthetic data, refers to artificially generated mock data used as a replacement for live data in testing scenarios. It is created to fill gaps in datasets and enhance the performance of machine learning models when actual data is unavailable or limited. It should be noted that the dummy data has similar characteristics to the live data that it is supposed to mimic, so that the accuracy and performance of the model remain relevant.

2.4.2 Overview of data augmentation

Data augmentation helps build relevant and useful machine learning models by decreasing the validation error according to the training error. An expanded dataset will encompass a wider range of potential data points, therefore reducing the gap between the training sets, validation sets, and any forthcoming test sets. Data augmentation assumes that additional information can be extracted from the original dataset through augmentations.

We will explore a manual form of data augmentation via the method of data warping. Data warping modifies existing images while keeping their labels preserved (Shorten & Khoshgoftaar, 2019), using techniques such as image rotation, warp shift, and Gaussian noise addition. We will analyse how data augmentation increases dataset diversity, leading to more generalised machine learning models, using data warping as an example.

We will use The MNIST database to illustrate data augmentation in image processing. The database contains 60,000 training and 10,000 test examples of handwritten digits.

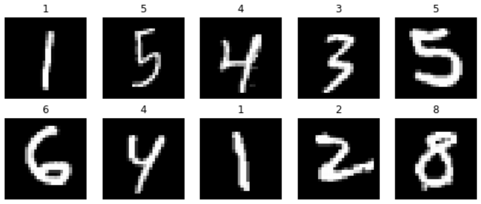


Figure 1: Some examples of handwritten digits available in the MNIST database.

Image rotation is a common utilised data augmentation technique in image processing, where an image is rotated around an axis. The safety and effectiveness of rotation augmentations depend heavily on the degree of rotation parameter used, and the ranges between -20^{o} and 20^{o} may prove useful for tasks like digit recognition. Increasing the degree of rotation excessively may no longer preserve the original label, which would reduce the usefulness of the augmentation (Shorten & Khoshgoftaar, 2019). In cases where the object is not centralised, shifting the image can add shift-invariance. Any empty space can be filled with a constant value like 0 or random noise (Sharma, 2019). Noise injection, which involves adding random values from Gaussian distribution, can help convolutional neural networks (CNNs) acquire more robust features. By using these respective augmentation techniques, the performance and generalisation of the machine learning model can be improved.

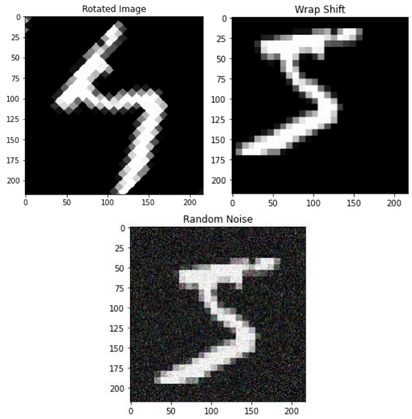


Figure 2: Rotation, wrap shift, and Gaussian noise addition

3. Data augmentation using machine learning

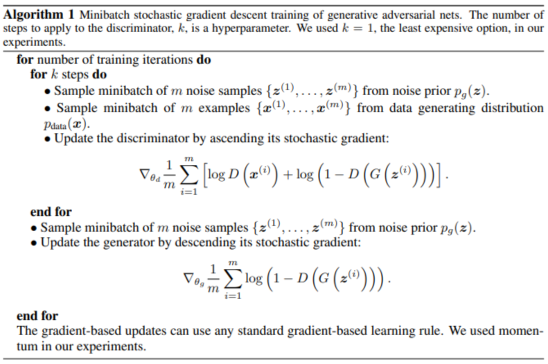
3.1 Generative Adversarial Networks (GANs)

3.1.1 Overview

GANs, introduced in 2014 by Ian Goodfellow and his collaborators, are a technique used for generative modelling, utilizing deep learning methodologies like CNNs to create new data samples (Brownlee, 2019) (Giles, 2018). GANs consist of two models: a generator and a discriminator. The generator captures the distribution of genuine examples for new data example generation. The discriminator, on the other hand, is typically a binary classifier, tasked with distinguishing between generated and real examples as accurately as it can. The optimisation of GANs represents a minimax optimisation problem, aiming to achieve Nash equilibrium at a saddle point (Gui et al., 2020). For the context where the generator and the discriminator are represented by multilayer perceptions, the fundamental structure is established with backpropagation, maximising the probability of assigning accurate labels to both training models and samples from the generator while also concurrently training it (Singh et al., 2021). The equation shown below in Figure 3 is the combined loss function, and the following algorithm in Figure 4 is used to train GANs, both derived by Goodfellow et al. in their original paper (Goodfellow et al., 2014).

$$\min_{G} \max_{D} V\,(D, G) = \mathbb{E}_{\chi\sim p\text{data}\,(x)}\,[\log \mathrm{D}(x)] + \mathbb{E}_{z\sim pz\,(z)}\,[\log(1 - \mathrm{D}(\mathrm{G}(z)))]$$

Figure 3: Combined loss function



```
Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, $k$, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations do
  for $k$ steps do
    • Sample minibatch of $m$ noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
    • Sample minibatch of $m$ examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
    • Update the discriminator by ascending its stochastic gradient:
        $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right].$
  end for
  • Sample minibatch of $m$ noise samples $\{z^{(1)}, \dots, z^{(m)}\}$ from noise prior $p_g(z)$.
  • Update the generator by descending its stochastic gradient:
        $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$
end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.
```

Figure 4: Algorithm used to train GANs using minibatch stochastic gradient descent.

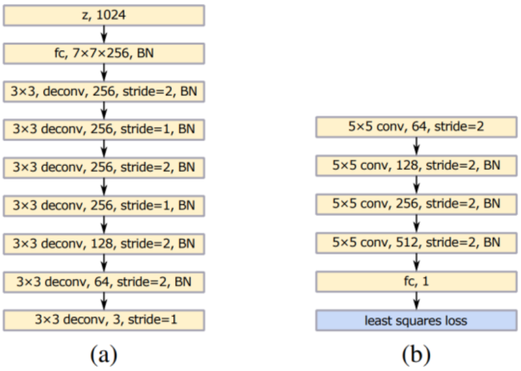

z, 1024
fc, 7×7×256, BN
3×3, deconv, 256, stride=2, BN
3×3 deconv, 256, stride=1, BN
3×3 deconv, 256, stride=2, BN
3×3 deconv, 256, stride=1, BN
3×3 deconv, 128, stride=2, BN
3×3 deconv, 64, stride=2, BN
3×3 deconv, 3, stride=1
5×5 conv, 64, stride=2
5×5 conv, 128, stride=2, BN
5×5 conv, 256, stride=2, BN
5×5 conv, 512, stride=2, BN
fc, 1
least squares loss
(a)
(b)

Figure 5: Model architecture of a GAN. "k x k conv/deconv, C, stride=S" represents a convolutional/deconvolutional layer with a (k x k) kernel, C output filters, and a stride of S. BN signifies that a batch normalisation layer follows the layer with BN. "fc, N" refers to a fully connected layer with N output nodes. The activation layers have been excluded for clarity. The left, (a), is the generator, while the right, (b), is the discriminator (Mao et al., 2017).

In the following example, we created a GAN using the MNIST dataset and programmed it with the PyTorch framework. The training images, shown in Figure 6, were chosen at random and weight initialisation was conducted for the neural network.

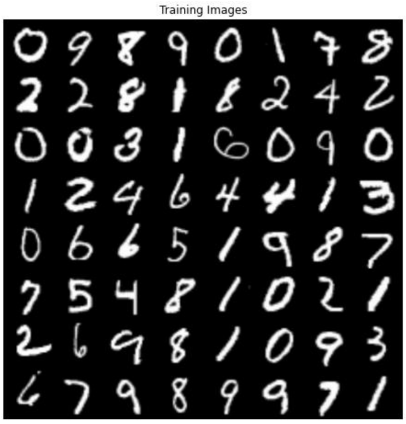


Figure 6: Training images used to create the GAN.

The generator and discriminator models were built, both consisting of 5 layers. A binary cross-entropy loss function was set up and used for the loss function, and ADAM optimiser was used for the optimiser. The GAN model was then trained. The process of training the discriminator involved giving the network labelled images coming from the generator, which are fake, and from the training data, which are real. The discriminator learns to classify real and fake images. Meanwhile, the generator's training aimed to improve the generation of fake images using feedback from the discriminator. After we trained the GAN for 5 epochs, we plot the real and fake images constructed by the GAN, as shown in Figures 7 and 8.

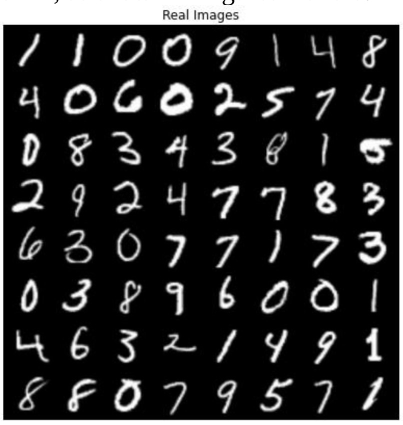


Figure 7: Real images used in training.

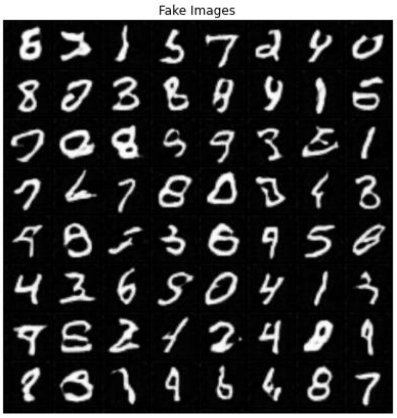


Figure 8: Fake images generated by the GAN.

3.2 Neural Style Transfer (NST)

3.2.1 Overview

NST, a technique used in deep learning and computer vision, combines the content of one image with the style of another. It represents a category of software algorithms that utilise neural networks to transform scenes or modify the environment of a media. Widely used in image and video editing software for image stylisation, it allows for faster creation of media and recreational use (Singh et al., 2021). NST generates new images that preserves the content, or structure, of the original image while incorporating the style of a second image.

3.2.2 Implementation of NST to augment data.

Here, we explore using NST to augment data in computer vision. We utilise two images, a style and a content image, and mix them together to produce an output image. Figure 9 shows the 2 images used.

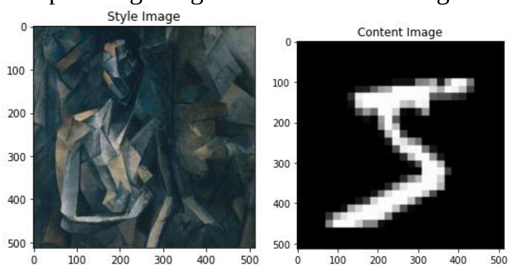


Figure 9: Style image and content image.

To implement NST, two loss functions are used: content loss and style loss. Content loss measures the distance between the features of the base image and that of the combination image, while ensuring the generated image is sufficient similar to the original one (Chollet, 2020). Style loss calculates the style difference. A pre-trained neural network is obtained from PyTorch's models, using only the 'features' module containing convolution and pooling layers. The L-BFGS algorithm from the PyTorch library is used to run our gradient descent to train the input image to minimise content and style losses. We then define a function to perform neural transfer, and the algorithm is executed, outputting an image as shown in Figure 10 (Jacq, n.d.).

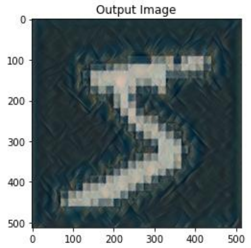


Figure 10: Output image of NST

4. Discussion

4.1 Overview

Dummy data generation using GANs and NST helps address data scarcity and has led to ground-breaking deep learning applications in various fields, including medical imaging, game development, image enhancements, and cybersecurity. GANs were amongst the most widely used generative AI techniques until the emergence of the Transformer in 2017.

Transformer models are a deep learning architecture that relies on attention, enabling much faster training and more effective understanding of context in sequential data than architectures that rely on recurrent or convolutional layers (Vaswani et al., 2017). Transformers are a fundamental technology and have given rise to various breakthroughs in large language models, such as Generative Pre-trained Transformers (GPTs), which are now being applied to multimodal AI tasks that efficiently correlate a wide range of content, surpassing even the capabilities of GANs (Lawton, 2023).

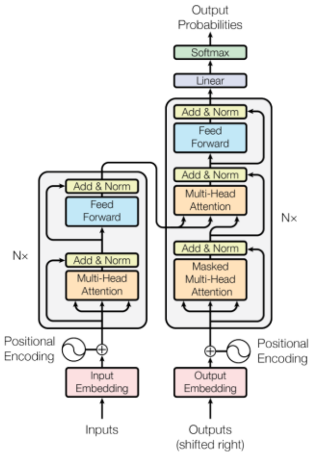


Figure 11: Model architecture of the Transformer.

The Transformer model employs an encoder-decoder architecture, as shown above, commonly used in neural sequence transduction models. In this structure, the encoder converts an input sequence of symbols into a continuous representation sequence, while the decoder generates an output sequence one symbol at a time, using previously generated symbols as additional input. The Transformer adheres to this overall architecture, using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder (Vaswani et al., 2017).

4.2 Benefits of using machine learning to augment data.

Using machine learning as a data augmentation technique offers several advantages over traditional and more manual methods. GANs, for example, require minimal supervision, continuously training themselves with their own generated data, and they can efficiently produce highly specialised data collections. This significantly reduces the need for manual or human labour in the data acquisition process (Lombardi, 2022).

4.3 Limitations of using machine learning to augment data.

Using machine learning as a data augmentation technique inevitably comes with some limitations. For instance, creating a GAN model is not so simple. It requires extensive technical expertise and sophisticated datasets, necessitating experienced developers for accurate model construction (Lombardi, 2022). In the case of NST, there is a risk of losing vital content information in the process of applying style transfer, which could negatively affect the performance of the model.

5. Conclusion

Performing dummy data generation and data augmentation using machine learning techniques like GANs and NSTs can be an efficient way of generating and augmenting data, especially when data acquisition is challenging. However, the emergence of transformers has been a game-changer in the field of AI development, and has led to significant advancements across numerous domains, including natural language processing, computer vision, and reinforcement learning. For instance, ChatGPT, a widely used AI language model now, exemplifies the transformer model's capabilities. It has even shown the ability to assist in text data augmentation (Dai et al., 2023).

It is probable that in the future, the large transformer models like GPT will also be commonly used for data augmentation techniques. However, there remains the concern of whether or not the data generated by these models are actually helping to improve the training performances of machine learning models with less error. Combining GANs and transformers may lead to more relevant and diverse data generation and augmentation, hence ensuring that the machine learning model's performance is not jeopardised.

This paper has provided an in-depth investigation on the reasons and impacts of data scarcity, and the different techniques used to generate dummy data and augment data using machine learning. We delved into the realm of deep learning where we explored the use of GANs and NST to generate synthetic data. Moreover, we discussed the potential of these models, together with transformers, in becoming more efficient data augmentation techniques in the future.

6. References

Abadi, A. (2021). *A survey on data-efficient algorithms in big data era.*

Brownlee, J. (2019). *A Gentle Introduction to Generative Adversarial Networks (GANs).*

Brownlee, J. (2021). *Weight initialization for deep learning neural networks.*

Burns, E. (2019). *What is machine learning and why is it important?*

Chollet, F. (2020). *Neural style transfer.*

Dai, H., Liu, Z., Liao, W., Huang, X., Cao, Y., Wu, Z., Zhao, L., Xu, S., Liu, W., Liu, N., Li S., Zhu, D., Cai, H., Sun, L., Li, Q., Shen, D., Liu, T., Li, X. (2023). *AugGPT: Leveraging ChatGPT for text data augmentation.*

Dimensional Research (2019). *Artificial Intelligence and Machine Learning Projects are obstructed by Data Issues.*

Dorfman, E. (2022). *How much data is required for machine learning?*

Giles, M. (2018). *The GANfather: The man who's given machines the gift of imagination.*

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014). *Generative Adversarial Nets.*

Gui, J., Sun, Z., Wen, Y., Tao, D., Ye, J. (2020) *A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications.*

IBM (n.d.). *What is underfitting?*

Jabbar, H. K., Khan, R. Z. (2015). *Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study).*

Jain, P., Gyanchandani, M., Khare, N. (2016). *Big data privacy: a technological perspective and view.*

JavaTpoint (n.d.). *Issues in Machine Learning.*

Lawton, G. (2023). *GAN vs. transformer models: Comparing architectures and uses.*

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998). *Gradient-based learning applied to document recognition.*

Lombardi, P. (2022). *What is GAN Machine Learning and what are its benefits?*

Mao, X., Li, Q., Xie, H., Lau, R. Y. K, Wang, Z., Smolley, S.P. (2017). *Least Squares Generative Adversarial Networks.*

MathWorks (n.d.). *What is Machine Learning?*

Noah (n.d.). *Dummy Data: Definition, Example & How to Generate It.*

Sambhav, K. (n.d.). *GANs and their applications.*

Sharma, P. (2019). *Image augmentation for deep learning using PyTorch – Feature engineering for images.*

Singh, A., Jaiswal, V., Joshi, G., Sanjeeve, A., Gite, S., Kotecha, K. (2021). *Neural Style Transfer: A Critical Review.*

Smolic, H. (2022). *How much data is needed for machine learning?*

Tableau (n.d.). *Real-world examples of machine learning (ML).*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). *Attention is all you need.*

Zhou, Z. (2018). *A brief introduction to weakly supervised learning.*